Guida alla Pulizia e Organizzazione - Fase CIK Matching

Data: 30 gennaio 2026
Versione: 1.0 - Post consolidamento MASTER

EXECUTIVE SUMMARY

È stato creato **04_cik_resolve_MASTER.R** che consolida tutte le strategie di CIK matching in un'unica pipeline robusta e ben documentata.

Questo file sostituisce completamente:

- 21_cik_resolve.R (v1)
- 21_cik_resolve_v2.R (v2)

I file vecchi vanno **archiviati** (NON eliminati) per tracciabilità storica.

I file nella cartella src/15_historical_cik/ vanno **mantenuti** come utility indipendenti.

PARTE 1: FILE NELLA CARTELLA src/20_resolve/

📁 File da MANTENERE (core pipeline)

✅ 00_run_pipeline.R - AGGIORNARE

Status: KEEP + UPDATE

Azione richiesta: Aggiornare la chiamata dello Step 2 da:

```
r

source("src/20_resolve/21_cik_resolve.R")
```

A:

```
r

source("src/20_resolve/04_cik_resolve_MASTER.R")
```

Modifiche specifiche:

```
r
```

```
# VECCHIO (linee 41-47)
cat("\n")
cat(paste(rep("=", 70), collapse = ""), "\n")
cat("EXECUTING STEP 2: CIK Resolution\n")
cat(paste(rep("=", 70), collapse = ""), "\n")

step2_start <- Sys.time()
source("src/20_resolve/21_cik_resolve.R")  # <-- CAMBIARE QUI
step2_time <- difftime(Sys.time(), step2_start, units = "mins")

# NUOVO
cat("\n")
cat(paste(rep("=", 70), collapse = ""), "\n")
cat("EXECUTING STEP 2: CIK Resolution (MASTER)\n")
cat(paste(rep("=", 70), collapse = ""), "\n")

step2_start <- Sys.time()
source("src/20_resolve/04_cik_resolve_MASTER.R")  # <-- NUOVA CHIAMATA
step2_time <- difftime(Sys.time(), step2_start, units = "mins")
```

Dopo la modifica, testare:

```
r

source("src/20_resolve/00_run_pipeline.R")
```

✅ 20_sample_apply.R - VERIFICARE

Status: KEEP (probabilmente ridondante con `03_apply_sample_restrictions.R`)

Domanda critica: Questo file è identico o simile a `src/20_clean/apply_sample_restrictions_v2.R`?

Azione richiesta:

1. Confrontare i due file:

```
r

  # Aprire entrambi e verificare:
  src/20_resolve/20_sample_apply.R
  src/20_clean/apply_sample_restrictions_v2.R
```

2. **Se identici:** Eliminare `20_sample_apply.R` e aggiornare `00_run_pipeline.R` per chiamare:

```
r

  source("src/20_clean/03_apply_sample_restrictions.R")
```

3. **Se diversi:** Documentare le differenze e decidere quale versione è corretta.

Motivazione: Evitare duplicazione logica e mantenere una sola "source of truth" per sample restrictions.

✅ 22_filing_identify.R - MANTENERE

Status: KEEP (fase successiva)

Nessuna azione immediata. Questo file sarà eseguito **DOPO** il CIK matching.

Dipendenze:

- Input: `data/interim/deals_with_cik.rds` (prodotto da `04_cik_resolve_MASTER.R`)
- Output: `data/interim/deals_with_filing.rds`

Quando eseguire: Solo dopo aver completato con successo il CIK matching e verificato coverage > 60%.

✅ README.md - AGGIORNARE

Status: KEEP + UPDATE

Modifiche richieste:

Sezione "Quick Start" (aggiornare Step 2):

````
markdown

## Quick Start

**Option 2: Run step by step**
```r
source("src/20_resolve/20_sample_apply.R")        # Step 1
source("src/20_resolve/04_cik_resolve_MASTER.R")  # Step 2 [AGGIORNATO]
source("src/20_resolve/22_filing_identify.R")     # Step 3
```
````

Nuova sezione da aggiungere: "CIK Resolution Strategies":

markdown

CIK Resolution Strategies (Step 2)

The MASTER pipeline (`04_cik_resolve_MASTER.R`) implements a four-strategy cascade:

Strategy 1: SEC Bulk Ticker File (~70% coverage)
- Direct lookup of active ticker symbols
- High precision, fast execution
- Source: https://www.sec.gov/files/company_tickers.json

Strategy 2: Historical Ticker Database (~20% recovery)
- Handles delisted/renamed companies
- Requires pre-built database (optional)
- Build with: `source("src/15_historical_cik/build_historical_ticker_database.R")`

Strategy 3: Browse-EDGAR Lookup (~5% recovery)
- Ticker or company name search via SEC browse interface
- Validates matches using submissions API
- Rate-limited (0.15s per request)

Strategy 4: Manual Review (remaining ~5%)
- Unmatched deals exported to `cik_unmatched.csv`
- Requires manual CIK lookup or exclusion decision

Expected Coverage
With all strategies: ****85-95% of deals****
- Strategy 1 alone typically achieves 65-75%
- Historical DB adds 15-20% for M&A samples (many delisted targets)
- Browse-EDGAR recovers edge cases

Configuration
Edit `CONFIG` list in `04_cik_resolve_MASTER.R`:
- `rate_limit_delay`: 0.15s (default, safe for SEC API)
- `name_similarity_min`: 0.75 (Jaro-Winkler threshold)
- `bulk_max_age_days`: 7 (cache refresh frequency)

🗂️ File da ARCHIVIARE (versioni vecchie)

📦 21_cik_resolve.R - ARCHIVIARE

Status: ARCHIVE (sostituito da MASTER)

Azione:

```bash
# Creare cartella archive se non esiste
mkdir -p src/20_resolve/_archive

# Spostare file
mv src/20_resolve/21_cik_resolve.R src/20_resolve/_archive/21_cik_resolve_v1_SUPERSEDED.R

# Aggiungere nota nel file
echo "# SUPERSEDED: Use src/20_resolve/04_cik_resolve_MASTER.R instead" | cat - src/20_resolve/_archive/21_cik_resc
```

Motivazione: Conservare per tracciabilità (codice originale), ma chiarire che è obsoleto.

📦 21_cik_resolve_v2.R - ARCHIVIARE

Status: ARCHIVE (sostituito da MASTER)

Azione:

```bash
mv src/20_resolve/21_cik_resolve_v2.R src/20_resolve/_archive/21_cik_resolve_v2_SUPERSEDED.R

echo "# SUPERSEDED: Use src/20_resolve/04_cik_resolve_MASTER.R instead" | cat - src/20_resolve/_archive/21_cik_resc
```

Motivazione: Stessa logica di v1 - conservare per storico.

PARTE 2: FILE NELLA CARTELLA `src/15_historical_cik/`

Questi file sono **utility indipendenti** che supportano Strategy 2 del MASTER pipeline.

📁 File da MANTENERE (tutti)

✅ build_historical_ticker_database.R - MANTENERE

Status: KEEP (utility essenziale)

Scopo: Costruisce il database storico ticker→CIK interrogando SEC Company Facts API.

Quando usare:

- **Prima volta:** Dopo aver completato ingestion e restrictions
- **Aggiornamenti:** Ogni 6-12 mesi (dataset storico cambia raramente)

Output: `data/interim/historical_ticker_cik_database.rds`

Tempo esecuzione: 2-4 ore (rate limiting SEC API)

Azione consigliata: Eseguire PRIMA di lanciare il MASTER pipeline per massimizzare coverage:

```
r

source("src/15_historical_cik/build_historical_ticker_database.R")
```

Nota: Il MASTER pipeline controlla automaticamente se questo file esiste. Se assente, Strategy 2 viene saltata (logged ma non bloccante).

✅ apply_historical_ticker_matching.R - MANTENERE (standalone)

Status: KEEP (utility diagnostica)

Scopo: Versione standalone del matching storico, utile per test/debug.

Quando usare:

- Testing isolato di Strategy 2
- Debug problemi di matching su subset specifico
- Analisi diagnostica coverage storico

NON necessario nella pipeline principale (Strategy 2 è già integrata nel MASTER).

Azione: Nessuna azione immediata. Lasciare disponibile come tool diagnostico.

✅ resolve_historical_tickers_complete.R - VERIFICARE

Status: KEEP o ARCHIVE (da determinare)

Domanda: Questo file duplica la logica già presente nel MASTER?

Azione richiesta:

1. Aprire il file e leggere header/commenti
2. Confrontare con `apply_historical_ticker_matching.R`
3. **Se identico/ridondante:** Archiviare in `src/15_historical_cik/Archive/`

4. **Se diverso:** Documentare la differenza nel README locale

Motivazione: Evitare confusione tra file con nomi simili.

PARTE 3: FILE NELLA CARTELLA `src/utils/`

Questi file sono **dipendenze core** del MASTER pipeline.

📁 File CRITICI (NON TOCCARE)

🔒 sec_http.R - CRITICO

Status: KEEP - DEPENDENCY

Usato da:

- `04_cik_resolve_MASTER.R`
- Tutti i file in `15_historical_cik/`

Funzioni chiave:

- `sec_user_agent()`: User-Agent per SEC API
- `sec_get()`: Rate-limited HTTP GET
- `cache_read()`, `cache_write()`: Gestione cache

Azione: NESSUNA. Non modificare senza testare tutti gli script dipendenti.

🔒 sec_cik_lookup.R - CRITICO

Status: KEEP - DEPENDENCY

Usato da:

- `04_cik_resolve_MASTER.R`

Funzioni chiave:

- `format_cik_10()`: Standardizzazione CIK (10 digit)
- `clean_ticker_for_lookup()`: Pulizia ticker (rimuove suffissi `.N`, `:US`)
- `clean_name_basic()`: Normalizzazione nomi aziendali
- `name_similarity_jw()`: Jaro-Winkler similarity
- `browse_edgar_lookup()`: Query Browse-EDGAR

- `sec_submissions_get()`: Query submissions JSON
- `score_candidate()`: Validazione match CIK
- `pick_best_cik()`: Selezione best match da candidati multipli

Azione: NESSUNA. Non modificare senza testare MASTER pipeline.

PARTE 4: STATO FINALE DELLA CARTELLA `src/20_resolve/`

Struttura target (dopo pulizia):

```
src/20_resolve/
├── 00_run_pipeline.R            [AGGIORNATO - chiama 04_cik_resolve_MASTER]
├── 20_sample_apply.R            [VERIFICARE se ridondante]
├── 04_cik_resolve_MASTER.R      [NUOVO - pipeline consolidata]
├── 22_filing_identify.R         [MANTENERE - fase successiva]
├── README.md                    [AGGIORNARE - documentare MASTER]
└── _archive/
    ├── 21_cik_resolve_v1_SUPERSEDED.R
    └── 21_cik_resolve_v2_SUPERSEDED.R
```

Struttura target `src/15_historical_cik/`:

```
src/15_historical_cik/
├── build_historical_ticker_database.R     [KEEP - utility builder]
├── apply_historical_ticker_matching.R     [KEEP - diagnostico]
├── resolve_historical_tickers_complete.R  [VERIFICARE ridondanza]
└── Archive/
    └── [vecchi tentativi]
```

PARTE 5: CHECKLIST AZIONI IMMEDIATE

Priority 1: Aggiornare pipeline orchestrator

- [ ] Aprire `src/20_resolve/00_run_pipeline.R`
- [ ] Cambiare chiamata Step 2 a `04_cik_resolve_MASTER.R`
- [ ] Testare esecuzione pipeline completa

Priority 2: Archiviare versioni vecchie

- [ ] Creare `src/20_resolve/_archive/`
- [ ] Spostare `21_cik_resolve.R` → `_archive/21_cik_resolve_v1_SUPERSEDED.R`
- [ ] Spostare `21_cik_resolve_v2.R` → `_archive/21_cik_resolve_v2_SUPERSEDED.R`
- [ ] Aggiungere header "SUPERSEDED" in entrambi

Priority 3: Verificare ridondanza

- [ ] Confrontare `20_sample_apply.R` con `../20_clean/apply_sample_restrictions_v2.R`
- [ ] Se identici, eliminare una versione e aggiornare `00_run_pipeline.R`
- [ ] Confrontare `resolve_historical_tickers_complete.R` con `apply_historical_ticker_matching.R`
- [ ] Se ridondanti, archiviare uno dei due

Priority 4: Aggiornare documentazione

- [ ] Aggiornare `src/20_resolve/README.md` (sezione Quick Start + Strategy details)
- [ ] Creare `src/15_historical_cik/README.md` se mancante
- [ ] Aggiornare README principale del progetto

Priority 5: Eseguire pipeline test

- [ ] (Opzionale) Eseguire `build_historical_ticker_database.R` per Strategy 2
- [ ] Eseguire `04_cik_resolve_MASTER.R` su sample ristretto (100 deals)
- [ ] Verificare output:
  - `deals_with_cik.rds` creato
  - `cik_diagnostics.csv` leggibile
  - `cik_resolution_log.txt` completo
- [ ] Controllare coverage (target: > 70% con Strategy 1 sola)

PARTE 6: COSA SUCCEDE SE...

❓ "Non ho tempo di costruire Historical DB ora"

Risposta: Nessun problema.

Il MASTER pipeline rileva automaticamente l'assenza del file `historical_ticker_cik_database.rds` e:

1. Logga un messaggio: "Historical ticker database not found - skipping Strategy 2"
2. Procede con Strategy 1 (SEC Bulk) e Strategy 3 (Browse-EDGAR)
3. Coverage attesa: 70-80% (accettabile per testing iniziale)

Puoi costruire il Historical DB in seguito e ri-eseguire il MASTER per aumentare coverage.

❓ "Coverage è < 60%, cosa faccio?"

Diagnostica:

1. Controlla `cik_resolution_log.txt` per errori API
2. Verifica connectivity SEC:

```r
httr::GET("https://www.sec.gov/files/company_tickers.json")
```

3. Controlla qualità ticker nel dataset:

```r
deals <- readRDS("data/processed/deals_restricted.rds")
sum(!is.na(deals$target_ticker)) / nrow(deals)  # % con ticker
```

Soluzioni:

- Se ticker quality < 80%: Problema upstream (ingestion o raw data)
- Se SEC API non risponde: Controllare firewall/proxy
- Se entrambi OK: Costruire Historical DB per recovery

❓ "Voglio testare solo Strategy 1 senza API calls"

Risposta: Modifica temporanea del CONFIG.

Nel file `04_cik_resolve_MASTER.R`, commenta le sezioni Strategy 3:

```r
# ============================================================================
# STRATEGY 3: BROWSE-EDGAR LOOKUP (TICKER/NAME SEARCH)
# ============================================================================

# log_strategy("STRATEGY 3: Browse-EDGAR Lookup")
# [commenta tutto il blocco]
```

Risultato: Solo SEC Bulk + Historical DB (nessuna API call lenta).

❓ "Alcuni match sembrano sbagliati (false positive)"

Diagnostica:

1. Controlla file `cik_diagnostics.csv`:

```csv
deal_id,target_name,target_ticker,target_cik,cik_source,cik_confidence,sec_company_name
123,ABC CORP,ABC,0000123456,sec_bulk,high,ABC CORPORATION
456,XYZ INC,XYZ,0000789012,browse_edgar,low,XYZ INDUSTRIES  # <-- Sospetto
```

2. Focus su righe con `cik_confidence = "low"` o `cik_source = "browse_edgar"`
3. Validazione manuale:
   - Cercare CIK su https://www.sec.gov/edgar/searchedgar/companysearch
   - Confrontare `target_name` vs `sec_company_name`

Correzione: Se confermi false positive:

1. Aggiungi CIK corretto in `cik_unmatched.csv` (formato: `deal_id,correct_cik`)
2. Crea script di override manuale (post-processing)

❓ "Voglio aumentare threshold di name similarity"

Risposta: Modifica CONFIG.

Nel file `04_cik_resolve_MASTER.R`, linea ~50:

```r
CONFIG <- list(
  # ...
  name_similarity_min = 0.75,  # Aumenta a 0.85 per essere più conservativo
  # ...
)
```

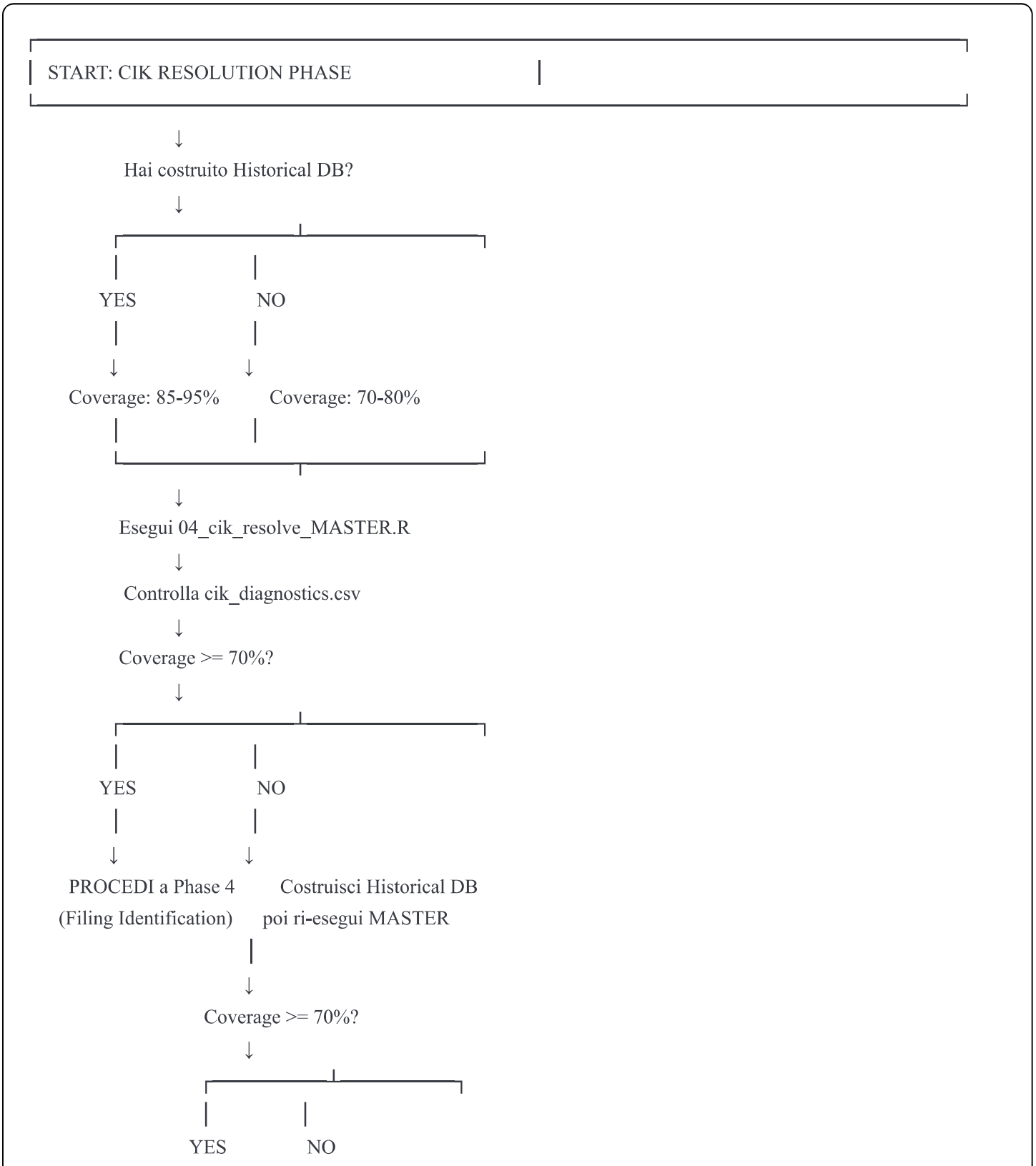
Trade-off:

- Threshold più alto (0.85) → Meno false positive, più unmatched

- Threshold più basso (0.70) → Più match, rischio false positive

Raccomandazione: Mantenere 0.75 per bilanciamento, validare manualmente low confidence.

PARTE 7: FLUSSO DECISIONALE CONSOLIDATO



```
        |           |
        ↓           ↓
    PROCEDI     Manual review
                cik_unmatched.csv
```

CONCLUSIONI

✅ File MASTER creato

`src/20_resolve/04_cik_resolve_MASTER.R` è pronto per l'uso e consolida tutte le strategie in un'unica pipeline tracciabile.

✅ Vecchie versioni identificate

`21_cik_resolve.R` e `21_cik_resolve_v2.R` vanno archiviati, non eliminati.

✅ Utilities protette

File in `src/utils/` sono dipendenze critiche e non vanno modificati.

✅ Historical DB opzionale ma raccomandato

Per sample M&A (molti target delisted), Historical DB è essenziale per 85%+ coverage.

✅ Pipeline testabile incrementalmente

Puoi eseguire Strategy 1 sola per test rapido, poi aggiungere Strategy 2 in seguito.

Prossimo step:

1. Archiviare vecchie versioni
2. Aggiornare `00_run_pipeline.R`
3. Eseguire `04_cik_resolve_MASTER.R` su sample test (100 deals)
4. Validare output e procedere a Filing Identification

Fine Guida